**The problem and how we solved it**

The problem we aimed to solve was discovering correlations between mental health factors, their strengths and the trends in which they follow. Our solution is a fully functional data solution that returns predicted data based off of it’s training.

**Datasets and sanitation methods**

The program is built on top of a free-open-source data set which is accessible through Kaggle.com.

This data set had 10,000 data points each containing 14 columns, some of which were not pertinent to our study. The information was generally clean, however for the purpose of our usage the first step was to drop the ID numbers from records in order to allow for order not to matter and make our analysis easier. Data was then read into a panda’s data frame and used for analysis by the model

For each model, we did not want it to consider the factor which we were studying immediately have access to the true value as this was the purpose our training. We would drop the desired columns from the and store them in a variable to reintroduce them after training for the model to draw a measure of its accuracy.

**Code**

The full code and supporting files can be accessed via:

<https://github.com/DakotaH5000/WGUCapstone>

**Reject or accept the hypothesis**

We must both accept and reject the hypothesis that was made about this program. It does not present the most accurate solution and the desired levels of confidence that we wish to see out of the model. This is due to faults in the underlying training data that can not be rectified without starting the process from scratch.

We however have created a model that if we choose to accept it, can make determinations as to the correlation strength of factors in determerming another factor. The model does output weighted strengths of factors (see data visualization section for example). This information could still provide key insights to users as to what factors should be focused on to ensure positive trends in mental health and which factors are less impactful and do not need such direct focus.

**Visualizations**

The model will give three primary types of visualizations/summaries to the end user. A output of the important information surrounding the results and it’s accuracy. A bar graph so users can visualize the weights of certain factors as determined by the model, and the overall placement of training data within the regression by the model. This allows the user to understand the result they are given and visualize the data that it was derived from.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A graph of a bar chart

Description automatically generated with medium confidence

**Is it accurate?**

The model is seemingly precise but in accurate. There were many questions about the validity of the data it was trained on. When viewed through tools utilized the data is too uniform and has no degree of randomness to be expected from such a large data set with as many data points.

This said, the accurate gave the same output repeatedly on similar inputs and is consistent in what information it does provide. The model could be better tuned with a more diverse data set or more data points but at the time of completion, it was not accurate.

**Testing**

Code was tested in multiple phases. A lot of the code was reusable between models so ensuring the boiler plate functioned properly was one of the most important first steps. Starting with the mental health model it was ensured the model could train and predict data. Taking this working boiler plate, regression models were created and implemented. These were tested and worked as desired.

The UI was built and tested, once again ensuring that each function worked properly with the mental health model, then integrating the regression models. These functions tested were: accepting a csv, accepting user input in the text box and the default input.

Finally the plotly integration was built and tested with the mental health model. Once it was delivering desirable results, it was implemented into the regression models and worked seamlessly after altering some strings for labeling axis.

**Solution file structure**

The file structure is:

WGUCapstone

DataModifiers

Encoders.py

inputValidation.py

Models

gamingHoursModel.py

mentalHealthModel.py

physicalActivityModel.py

screenTimeModel.py

sleepHoursModel.py

socialMediaModel.py

techHoursModel.py

trainingData

testData.csv

mental\_health\_and\_technology\_usage\_2024.csv

Main.py

**Solution QuickStart guide**